HW3 Sol

1. i. First prove that if a set $S$ of natural numbers is decidable, then it can be enumerated in increasing order. If $S$ is decidable there is a machine $M$ which decides it. Using this machine, we construct the enumerator $E$ which enumerates $S$ in increasing order.

$E =$ “Ignore the input.
1. Repeat the following for $i = 1, 2, 3, \ldots$
2. Run $M$ on $i$. If $M$ accepts, print $i$.”

Because $M$ is deciding $S$, $M$ is guaranteed to halt. If $M$ accepts $i$, then $i$ is in the set and should be printed. Otherwise, $E$ just moves on to the next natural number. The output will be in increasing order because we check for membership in $S$ in increasing order.

Now prove that if $S$ can be enumerated in increasing order, then $S$ is decidable. There are two cases for this proof: $S$ is finite and $S$ is infinite. If $S$ is finite, $S$ can be decided just by hardcoding the acceptance of each element of $S$ into a TM. So, we worry only about the case where $S$ is infinite. Let $E'$ be the machine which enumerates $S$ in increasing order. From this machine, we build a machine $M'$ which decides $S$.

$M' =$ “On input $w$:
1. Run $E'$.
2. For each output $x$ that $E'$ prints:
   3. Check if $x \geq w$.

Since $S$ is infinite eventually $x \geq w$ will be true. It this point we halt $E'$s enumeration and we got to step 4.

4. If $x = w$, accept as we know that $x \in S$. If $x \not= w$, reject as we know that $E'$ will never enumerate $x$.

ii. Yes. If the set can be enumerated in non-increasing order it must be finite. Every element in the set must be smaller than the first element that gets enumerated and there are only finitely many numbers between this first number and 0 which can be enumerated. Every finite set is decidable, so every set of natural numbers which can be enumerated in non-increasing order is decidable.

2. i. False. The Church/Turing thesis says that a TM can compute any algorithm but makes no claim about being able to decide all recognizable languages.

ii. True. For example, take the set $\{n\}$ for each $n \in \mathbb{N}$. Each of these is decidable and there are infinitely many of them.

iii. False. A 1 tape TM can simulate a 3 tape TM, so if the language can be decided on a 3 tape TM it can also be decided (and thus recognized) on the 1 tape TM.
iv. False. If the TM loops on any inputs it does not decide a language, it only recognizes one.

3. Since $D$ is decidable there is some machine $M$ which decides it. To prove that $S$ is recognizable we design the machine $M'$ which recognizes it.

$M' = \text{"On input } x:\$
1. Repeat the following for $i=1, 2, 3, ...$
   2. Run $M$ on $(x, i)$. If $M$ accepts, accept.$"$

Since $M$ decides $D$, running $M$ on $(x, i)$ is guaranteed to halt. If $M$ accepts, $(x, i)$ is in $D$ and thus $x \in S$, so $M'$ should accept $x$. If there does not exist an $i$ for which $(x, i) \in D$, $M'$ will loop forever but $x \not\in S$ so $M'$ is allowed to loop on $x$.

4. There are two ways that a TM which can’t move left can loop: it can stay on the same tape square forever or it can move right forever.

The Main Fact we need is: If the TM is in state $q$ and the head is on symbol $s$, then the machine is in a loop where it stays forever if it stays on the same tape cell, sees $s$ again and is in state $q$ again. So if the machine stays on a tape cell for $|Q||\Gamma| + 1$ steps in a row, it has to have repeated a state and a symbol since there are only $|Q||\Gamma|$ combinations of states and symbols. Thus, if the TM stays on one cell for $|Q||\Gamma| + 1$ steps, the TM is in a loop.

So now, use the Main Fact to describe how to determine if a TM $M$ which never moves left eventually halts or loops on an input string $w$.

Step 1: For each character $c$ in the string $w$ that $M$ reaches in its computation we count how many steps it takes as it computes before it either halts or moves to the right. As the Main fact tells us, if the counter never reaches $|Q||\Gamma| + 1$ steps on that square, the computation either halts on the square or moves to the right. Otherwise we stop knowing $M(w)$ never halts on $w$. We do this for every square of $w$.

Step 2: If $M(w)$ does not halt in step 1 then we know $M(w)$ eventually reaches the blank squares, then we repeat step 1 on the next $|Q| + 1$ B squares. If $M(w)$ halts during this time we know it does so.

Otherwise $M(w)$ must loop because for two blank symbols the state that $M$ is in when reaching them are the same.

Finally, we can now decide $M(w)$.

To do this we use the above steps to decide if $M(w)$ halts.

If it does not halt we reject, otherwise we run $M(w)$ until it halts and do what it does.